EFFICIENT IBE-PKE PROXY RE-ENCRYPTION

Takeo Mizuno[1,2] and Hiroshi Doi[2]
[1]*NTT Data Corporation, 3-3-3 Toyosu, Koutou-ku, Tokyo, Japan*
[2]*Institute of Information Security, 2-14-1 Tsuruya-cho, Kanagawa-ku, Yokohama-shi, Kanagawa, Japan*

Keywords: Proxy re-encryption, public key encryption, identity-based encryption, bilinear maps.

Abstract: In proxy re-encryption schemes, a semi-trusted entity called proxy can convert a ciphertext encrypted for Alice into a new ciphertext for Bob without seeing the underlying plaintext. Several proxy re-encryption schemes have been proposed, however, only one scheme which enables the conversion of IBE ciphertexts to PKE ciphertexts has been proposed and it has some drawbacks. In that scheme, the size of the re-encrypted ciphertext increases and Bob must be aware of existence of the proxy, which means Bob cannot decrypt a re-encrypted ciphertext with same PKE decryption algorithm.

We propose a new, efficient scheme that enables the conversion of IBE ciphertexts to PKE ciphertexts, and prove CPA security in the standard model. In our scheme, the size of the re-encrypted ciphertext is optimal and Bob does not aware of existence of the proxy. As far as we knows, this is the first IBE-PKE type scheme that holds the above properties.

1 INTRODUCTION

In proxy re-encryption schemes, a semi-trusted entity called proxy can convert a ciphertext encrypted for Alice into a new ciphertext, which another user Bob can decrypt with his own secret information without revealing the underlying plaintext. The proxy is not fully trusted, i.e., the proxy cannot reveal Alice's or Bob's secret key, and can not learn the plaintext during the conversion.

There are many useful applications of these schemes. For instance, Alice can securely forward encrypted e-mails to Bob in her absence.

The proxy converts the messages which encrypted under the email address alice@foo.com into another ciphertexts encrypted under bob@foo.com. The proxy does not learn the content of the messages during conversion and Alice can forward message without revealing her secret key.

Several proxy re-encryption schemes have been proposed in the context of public key encryption (PKE), e.g., ElGamal or RSA. Other schemes have been proposed in the context of Identity Based Encryption (IBE) which the sender encrypts a plaintext using arbitral strings that represents the recipient's identity as the public key. The IBE has proven useful in solving public key-distribution issues of traditional certificate based PKE schemes.

Matsuo proposed two proxy re-encryption schemes. The former one enables conversion between IBE users and the latter one enable the conversion of PKE ciphertexts to IBE ciphertexts in (T.Matsuo, 2007).

The latter one called hybrid scheme can be useful in PKE and IBE mixed environments. Matsuo also classify proxy re-encryption schemes as follows:

[PKE-PKE]-Type Scheme. Proxy converts PKE ciphertexts to PKE ciphertexts.(M.Mambo and E.Okamoto, 1997), (M.Blaze et al., 1998), (M.Jakobsson, 1999), (Y.Dodis and A.Ivan, 2003), (L.Zbou et al., 2004), (G.Ateniese et al., 2005), and (R.Canetti and S.Hohenberger, 2007) have been proposed as this type.

[IBE-IBE]-Type Scheme. Proxy converts IBE ciphertexts to IBE ciphertexts. (Y.Dodis and A.Ivan, 2003), (T.Matsuo, 2007), and (M.Green and G.Ateniese, 2007) have been proposed as this type.

[PKE-IBE]-Type Scheme. Proxy converts PKE ciphertexts to IBE ciphertexts. (T.Matsuo, 2007) has been proposed as this type.

[IBE-PKE]-Type Scheme. Proxy converts IBE ciphertexts to PKE ciphertexts. (M.Green and

Mizuno T. and Doi H. (2008).
EFFICIENT IBE-PKE PROXY RE-ENCRYPTION.
In *Proceedings of the International Conference on Security and Cryptography*, pages 285-293
DOI: 10.5220/0001917802850293

G.Ateniese, 2007) has been proposed as this type.

Green and Ateniese proposed the [IBE-PKE]-type scheme in (M.Green and G.Ateniese, 2007); however their scheme has following drawbacks.

1. The size of the re-encrypted ciphertext increases as compared to that of the original ciphertext.
2. The decryption algorithm of the re-encrypted ciphertext is different from the original decryption of the PKE scheme.

[IBE-IBE] type and [PKE-PKE] type of proxy re-encryption schemes have been proposed without such drawbacks. One of the theoretical interests is to construct the [IBE-PKE]-type proxy re-encryption scheme which does not have such drawbacks.

1.1 Entities of Proxy Re-Encryption

Generally, proxy re-encryption schemes have the following entities.

Sender. This entity encrypts plaintexts using a delegator's public key.

Delegator. This entity possesses the secret key corresponding to the public key used by the sender, and delegates decryption rights.

Delegatee. The decryption rights delegates to this entity from the delegator. The delegatee can decrypt re-encrypted ciphertexts own secret key, and without the delegator's secret key.

Proxy. This semi-trusted entity re-encrypts ciphertexts with a re-encryption keys, and outputs the ciphertexts, which the delegatee can decrypt using his own secret key without revealing underlying the plaintexts.

In [IBE-IBE], [IBE-PKE] and [PKE-IBE] type schemes have an additional entity PKG (Private Key Generator), which generates IBE secret keys. In our schemes this trusted entity take a part of re-encryption key generation.

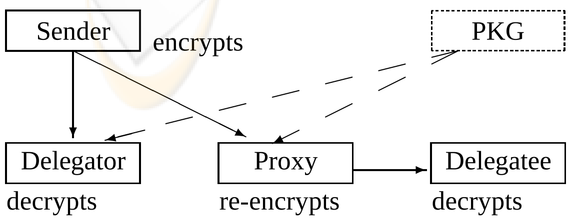

Figure 1: Entities of proxy re-encryption.

1.2 Security of Proxy Re-Encryption

With regard to the security of proxy re-encryption schemes Green and Ateniese pointed out the previous schemes achieve a security only for chosen plaintext attacks (CPA), and also proposed a new scheme achieves chosen ciphertext attacks (CCA) security in (M.Green and G.Ateniese, 2007).

Green and Ateniese described that in the previous schemes, proxy cannot verify ciphertexts and grant adversaries invalid re-encryption. Hence, malicious delegatee can use a re-encryption oracle as a decryption oracle. Furthermore they proposed CCA-secure scheme with random oracle model using Canetti, Halevi and Kats (CHK) (R.Canetti et al., 2004) technique, which enables the proxy to validate ciphertexts.

After Green and Ateniese pointed out the security problems with the previous schemes, Canetti and Hohenberger proposed CCA-secure [PKE-PKE]-type Re-Encryption scheme in the standard model (R.Canetti and S.Hohenberger, 2007).

In this paper, we propose a new [IBE-PKE]-type scheme, which achieves CPA-security only. However it might be possible achieve CCA-security using Green and Ateniese technique in (M.Green and G.Ateniese, 2007).

1.3 Our Contribution

We propose the first [IBE-PKE]-type proxy re-encryption scheme, which holds the following advantages simultaneously.

- Our scheme achieves optimal ciphetext size. The size of a re-encrypted ciphertext is same as a PKE ciphertext, while (M.Green and G.Ateniese, 2007) [IBE-PKE]-type scheme requires additional elements of ciphertext to support re-encryption.
- Our scheme achives proxy invisibility which means delegatee does not require additional algorithm for decryption of a re-encrypted ciphertext. The delegatee can decrypt ciphertexts without being aware of the existence of the proxy, while it is required in (M.Green and G.Ateniese, 2007).
- Our scheme is selective-ID secure in the standard model, while previous [IBE-PKE]-type scheme in (M.Green and G.Ateniese, 2007) might be full-ID secure in the random oracle model. Furthermore our scheme might be possible to extend full-ID secure using IBE proposed in (B.Waters, 2005).
- In Our scheme the PKG generates re-encryption keys, while (M.Green and G.Ateniese, 2007) del-

egator generates re-encryption keys himself individually. However this property should not affect security of our scheme, because the PKG is a trusted entity in the IBE schemes, and does not generate re-encryption key without notifying the delegator.

1.4 Organisation

The rest of paper consists of 4 sections. In Sec. 2 gives some definitions and preliminaries. In Sec. 3 we define security of IBE-PKE type proxy re-encryption. In Sec. 4 we present the IBE-PKE type proxy re-encryption scheme, and finally conclude this study in Sec. 5.

2 PRELIMINARIES

In this section, We describe the settings and computational assumptions used in this paper. We then define an [IBE-PKE]-type proxy re-encryption scheme and its security.

2.1 Bilinear Groups

Let $\mathbb{G}$ and $\mathbb{G}_1$ be the two multiplicative cyclic groups of prime order p, and g be a generator of $\mathbb{G}$. We say that $\mathbb{G}_1$ has an admissible bilinear map $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_1$ if the following conditions hold.

1. $\hat{e}(g^a, g^b) = \hat{e}(g,g)^{ab}$ for all a,b
2. $\hat{e}(g,g) \neq 1$

We say that $\mathbb{G}$ is a bilinear group if the group action in $\mathbb{G}$ can be computed efficiently and there exists a group $\mathbb{G}_1$ and an efficiently computable bilinear map $\hat{e}$ as above.

2.2 Decisional Bilinear Diffie-Hellman Assumption (dBDH)

The dBDH problem (D.Boneh and X.Boyen, 2004) in $\mathbb{G}$ as follows: Let $\mathbb{G}$ be a bilinear group of prime order p with an efficiently computable pairing $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_1$, let g be a random generator of $\mathbb{G}$. The dBDH problem is to decide, given a tuple $g, g^a, g^b, g^c, T \in \mathbb{G}^4 \times \mathbb{G}_1$ as inputs, (where $a,b,c \in_R \mathbb{Z}_p^*$), whether $T = \hat{e}(g,g)^{abc}$ or if T is a random element of $\mathbb{G}_1$.

Let k be a security parameter of suffcient size, we define the advantage of an algorithm $\mathcal{A}$ as follows:

$$Adv_{\mathcal{A}}^{dBDH} = |\Pr[\mathcal{A}(g, g^a, g^b, g^c, \hat{e}(g,g)^{abc}) = 0] \quad - \Pr[\mathcal{A}(g, g^a, g^b, g^c, T) = 0]|$$

where the probability is taken over the random choice of the generator g,the random choice of a,b,c in $\mathbb{Z}_p^*$, the random choice of T in $\mathbb{G}_1$, and the random bits consumed by $\mathcal{A}$. We say that (k,t,ε)-dBDH assumption holds in $\mathbb{G}$ if no t-time algorithm has advantage $Adv_{\mathcal{A}}^{dBDH} < \varepsilon$ $\mathbb{G}$ under security parameter k.

2.3 Identity Based Encryption Scheme

Identity Based Encryption (IBE) consists of the following algorithm.

$\mathbf{SetUp_{IBE}(k)}$. Given a security parameter k as input, a trusted entity Private Key Generator (PKG) generates a master key *mk* and public parameters *params*, and outputs *mk* and *params*.

$\mathbf{KeyGen_{IBE}(mk, params, ID)}$. For inputs of a master key *mk*, public parameters *params*, and an identity *ID*, the PKG outputs a IBE secret key sk_{ID} corresponding to the identity.

$\mathbf{Enc_{IBE}(ID, params, M)}$. For inputs of an identity *ID*, public parameters *params*, and a plaintext M, computes an IBE ciphertext C_{IBE}

$\mathbf{Dec_{IBE}(sk_{ID}, params, C_{IBE})}$. For inputs of a IBE secret key sk_{ID}, public parameters *params*, and an IBE ciphertext C_{IBE}, decrypts and outputs a plaintext M.

2.4 Public Key Encryption Scheme

Public Key Encryption (PKE) consists from following algorithms.

$\mathbf{KeyGen_{PKE}(k, params)}$. Given a security parameter k and IBE public parameters *params* as input, outputs PKE key pair $\langle SK, PK \rangle$ where SK is a secret key, PK is the corresponding public key PK.

$\mathbf{Enc_{PKE}(PK, M, params)}$. For inputs of a public key PK and plaintext M, IBE public parameters *params*, outputs the PKE ciphertext C_{PKE}.

$\mathbf{Dec_{PKE}(SK, C_{PKE}, params)}$. For inputs a secret key *sk*, PKE ciphertext C_{PKE}, and IBE public para metes *params*, decrypts and outputs a plaintext M.

2.5 IBE-PKE Proxy Re-Encryption Scheme

[IBE-PKE]-type proxy re-encryption (IBE-PKE-PRE) consists of the following algorithm

$\mathbf{KeyGen_{PRO}(mk, ID, PK, PK_R, params)}$. For inputs of a master key *mk*, a delegator's identity *ID*, delegatee's PKE public key PK and public key for Re-Encryption PK_R, and IBE public parameters

Table 1: comparison of [IBE-PKE] type scheme.

Property	(M.Green and G.Ateniese, 2007)	This work
Optimal size of re-encrypted ciphertext	No	Yes
Proxy Invisible	No	Yes
Re-encryption key generator	Delegator	PKG

params, a re-encrypt key $rk_{ID\rightarrow PKE}$ is output to the proxy.

$\mathbf{ReEnc_{PRO}}(\mathbf{ID}, \mathbf{rk_{ID\rightarrow PKE}}, \mathbf{params}, \mathbf{C_{IBE}})$. For inputs of a delegator's identity ID, a re-encrypt key $rk_{ID\rightarrow PKE}$, IBE public parameters *params*, and a IBE ciphertext C_{IBE}, the proxy re-encrypts and outputs a PKE ciphertext C_{PKE} to the delegatee.

3 CHOSEN PLAINTEXT SECURITY FOR IBE-PKE PROXY RE-ENCRYPTION

We define chosen plaintext security for the [IBE-PKE]-type scheme according to the following game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. We define two types of attacks, an adversary attacks against the IBE scheme and another against the PKE scheme. Hence, in the following game, we define an adversary attacks against the IBE scheme as $(TYPE = IBE)$ and an adversary attacks against the PKE as $(TYPE = PKE)$.

We design the following game on the basis of Boneh and Boyen's selective ID secure IBE game (D.Boneh and X.Boyen, 2004) and Green and Ateniese's proxy re-encryption game (M.Green and G.Ateniese, 2007). We show even if an adversary obtains additional informations related to proxy re-encryption, such as re-encryption keys, it does not make the underlying IBE or PKE schemes weak.

In the following game, the adversary is allowed to adaptively conduct IBE secret key queries, PKE secret key queries and re-encryption key queries. These queries imply the following situation that: The adversary corrupts IBE users to obtain their IBE secret keys, corrupts PKE users to obtain their PKE secret keys and corrupts the proxy to obtain re-encryption keys. We classify PKE users under two party, *honest* party and *corrupted* party by adversary. The adversary can obtain a PKE secret key of a *corrupted* party, but restricted to get re-encryption keys which can convert an IBE ciphertext corresponding to *target* identity to a PKE ciphertext for the *corrupted* party, because the adversary obviously wins the game. The adversary also restricted to obtain a PKE secret key of a *honest* party, but does not restricted to get re-encryption keys which can convert an IBE ciphertext to a PKE ciphertext for the *honest* party.

Definition 3.1. *(Security of [IBE-PKE]-type proxy re-encryption) Let $\mathcal{S}$ be an IBE-PKE-PRE scheme defined as a tuple of algorithms $(Setup_{IBE}, KeyGen_{IBE}, Enc_{IBE}, Dec_{IBE}, KeyGen_{PKE}, Enc_{PKE}, Dec_{PKE}, KeyGen_{PRO}, ReEnc_{PRO})$. The security is defined according to the following game, where $TYPE \in \{IBE, PKE\}$.*

Initialization. If the adversary $\mathcal{A}$ is $(TYPE = IBE)$, $\mathcal{A}$ outputs a target identity ID^*.

SetUp. The challenger $\mathcal{C}$ generates $params, mk$ by running $\mathbf{SetUp_{IBE}}$. $\mathcal{C}$ also generates PKE keys $\langle PKE_j, PK_j, PK_{R_j}, SK_j\rangle$ where PKE_j is a PKE user identity ,PK_j and SK_j are PKE key pairs, PK_{R_j} is a public key for re-encryption corresponding to PKE_j, $\mathcal{C}$ placed them in lists:

PPKL (PKE Public Key List) Holds PKE user identities PKE_j, PKE public keys PK_j and PKE public keys for re-encryption PK_{R_j}.

PSKL (PKE Secret Key List) Holds PKE user identities PKE_j, PKE secret keys SK_j and *mark* which holds a flag that PKE user is a *honest* party or *corrupted* party by $\mathcal{A}$.

Then, $\mathcal{C}$ gives $\langle params, PPKL\rangle$ to $\mathcal{A}$, and keep $\langle mk, PSKL\rangle$ secret to it self.

Phase 1. Given $\langle params, PPKL\rangle$, $\mathcal{A}$ adaptively queries $\mathcal{C}$. $\mathcal{C}$ responds as follows:

$\mathbf{Extract_{IBE}}(\mathbf{ID_i})$. $\mathcal{A}$ queries the IBE user's secret key sk_{ID_i} with an identity ID_i where $ID_i \neq ID^*$. $\mathcal{C}$ responds sk_{ID_i} corresponding to ID_i to $\mathcal{A}$.

$\mathbf{Extract_{IBE\rightarrow PKE}}(\mathbf{ID_i}, \mathbf{PKE_j})$. $\mathcal{A}$ queries the re-encryption key $rk_{ID_i\rightarrow PKE_j}$ with an identity ID_i and a PKE user identity PKE_j. $\mathcal{C}$ responds $rk_{ID_i\rightarrow PKE_j}$ corresponding to ID_i and PKE_j to $\mathcal{A}$.

$\mathbf{Extract_{PKE}}(\mathbf{PKE_j})$. $\mathcal{A}$ queries the PKE secret key SK_j with a PKE user identity PKE_j. $\mathcal{C}$ responds SK_j corresponding to PKE_j to $\mathcal{A}$.

Challenge. After Phase 1 , $\mathcal{A}$ outputs two equal length plaintexts M_0, M_1 and sends them to $\mathcal{C}$. $\mathcal{C}$ picks $b \in_R \{0,1\}$ and encrypts M_b.

If $(TYPE = IBE)$, $\mathcal{C}$ encrypts M under an identity ID^* and responds C^*_{IBE} to $\mathcal{A}$.

If $(TYPE = PKE)$, $\mathcal{A}$ selects a target PKE user identity PKE^* from *honest* parties, and also sends it to $\mathcal{C}$. $\mathcal{C}$ encrypt M under an PKE user identity PKE^* and responds C^*_{PKE} to $\mathcal{A}$.

Phase 2. $\mathcal{A}$ continues with the queries as in **Phase 1**, and $\mathcal{C}$ responds as before.

Solve. Finally $\mathcal{A}$ outputs a guess result $b' \in \{0,1\}$.

The adversary $\mathcal{A}$ wins if $b' = b$.

Besides the above game, during Phase 1 and Phase 2, $\mathcal{A}$ restricts the following queries which $\mathcal{A}$ can decrypt a challenge ciphertext only using $\mathcal{C}$'s answers.

If $(TYPE = IBE)$, the following queries are restricted.

- $\mathbf{Extract_{IBE}}(\mathbf{ID}^*)$, where ID^* is the challenge identity.
- $\mathbf{Extract_{PKE}}(\mathbf{PKE_j})$, where PKE_j is a *honest* party's identity.
- $\mathbf{Extract_{IBE \to PKE}}(\mathbf{ID}^*, \mathbf{PKE_j})$, where ID^* is the challenge identity and PKE_j is a *corrupted* party's PKE user identity.

If $(TYPE = PKE)$, the following queries are restricted.

- $\mathbf{Extract_{PKE}}(\mathbf{PKE_j})$, where PKE_j is a *honest* party's PKE user identity.

Definition 3.2. *Let $\mathcal{A}$ be an adversary against IBE-PKE-PRE. Define the IND-sPr-CPA advantage of $\mathcal{A}$ as follows:*

$$Adv^S_{\mathcal{A}}(k) = 2(\Pr[b = b'] - 1/2).$$

We say that the IBE-PKE-PRE scheme is (k,t,q,ε) adaptive chosen plaintext secure if for any t-time adversary $\mathcal{A}$ that makes at most q chosen queries under a security parameter k, we have that $Adv^S_{\mathcal{A}}(k) < \varepsilon$.

4 EFFICIENT IBE-PKE TYPE PROXY RE-ENCRYPTION

We construct an [IBE-PKE]-type proxy re-encryption scheme (IBE-PKE-PRE) which achieves CPA-secure without Random Oracle.

IBE-PKE-PRE is enable conversion of an IBE ciphertext to a PKE ciphertext. Our scheme IBE-PKE-PRE uses Boneh and Boyen's selective ID secure IBE scheme (D.Boneh and X.Boyen, 2004) (BB-IBE) for IBE scheme. We construct a new (but very similar to PKE scheme proposed in (G.Ateniese et al., 2005)) ElGamal-type PKE scheme for IBE-PKE-PRE and propose a re-encryption scheme that converts a BB-IBE ciphertext to this PKE scheme's ciphertext.

4.1 BB-IBE Scheme

$\mathbf{SetUp_{IBE}(k)}$. Given security parameter k as input, let $\mathbb{G}, \mathbb{G}_1$ be a bilinear group of prime order p. Let $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ be the bilinear map. Select a random generator $g \in \mathbb{G}$ and random elements $h, g_2 \in \mathbb{G}$. Pick a random element $\alpha \in \mathbb{Z}^*_p$ and set $g_1 = g^\alpha, mk = \alpha$ and set $params = \langle g, g_1, g_2, h \rangle$.

Let mk be a master secret key, and $params$ be the public parameters.

$\mathbf{KeyGen_{IBE}(mk, params, ID)}$. Given master secret key $mk = \alpha$, public parameters $params$ and an identity ID as input, the PKG picks a random element $u \in \mathbb{Z}^*_p$ and outputs an IBE secret key sk_{ID}.

$$sk_{ID} = \langle d_1, d_2 \rangle = \langle g_2^\alpha \left(g_1^{ID} h\right)^u, g^u \rangle$$

$\mathbf{Enc_{IBE}(ID, params, M)}$. Given an identity ID, public parameter $params$ and plaintext $M \in \mathbb{G}_1$ as input, select a random element $r \in \mathbb{Z}^*_p$ and output an IBE ciphertext C_{IBE}.

$$C_{IBE} = \langle C_1, C_2, C_3 \rangle = \left\langle g^r, \left(g_1^{ID} h\right)^r, M\hat{e}(g_1, g_2)^r \right\rangle$$

$\mathbf{Dec_{IBE}(sk_{ID}, params, C_{IBE})}$. Given an IBE secret key sk_{ID}, public parameters $params$ and an IBE ciphertext C_{IBE} as input, output a plaintext M.

$$M = \frac{C_3 \hat{e}(d_2, C_2)}{\hat{e}(d_1, C_1)}$$

4.2 PKE Scheme

$\mathbf{KeyGen_{PKE}(k, params)}$. Given security parameter k and BB-IBE public parameters $params$ as input, select a random element $x \in_R \mathbb{Z}^*_p$ and set $SK = x$, $PK = g^x$, output SK as a PKE secret key and PK as a PKE public key.

If PKE user accepts delegation, PKE user also publish public key for re-encryption $PK_R = g_2^{1/SK}$. If PKE user does not wish to accept delegation, PKE user does not publish public key for re-encryption value.

$\mathbf{Enc_{PKE}(PK, M, params)}$. Given a PKE public key PK, a plaintext $M \in \mathbb{G}_1$ and BB-IBE public parameters $params$ as input, pick a random element $v \in \mathbb{Z}^*_p$ and output a PKE ciphertext $C_{PKE} = \langle X, Y \rangle$.

$$C_{PKE} = \langle X, Y \rangle = \langle \hat{e}(g,g)^v, M \cdot \hat{e}(g, PK)^v \rangle$$

$\mathbf{Dec_{PKE}}(\mathbf{SK}, \mathbf{C_{PKE}}, \mathbf{params})$. Given a PKE secret key SK, a PKE ciphertext C_{PKE} and BB-IBE public parameters *params* as input, output a plaintext M.

$$M = Y/X^{SK}$$

4.3 Proxy Re-Encryption

$\mathbf{KeyGen_{PRO}}(\mathbf{mk}, \mathbf{ID}, \mathbf{PK}, \mathbf{PK_R}, \mathbf{params})$. Given a master secret key $mk = \alpha$, a delegator's identity ID and a delegatee's PKE public key PK and public key for re-encryption PK_R as input, PKG outputs a re-encryption key $rk_{ID\to PK} = \langle rk_1, rk_2\rangle$ or $\perp$.

1. If $\hat{e}(PK, PK_R) \neq \hat{e}(g_2, g)$, then output $\perp$ and halt.
2. Compute $rk_{ID\to PK}$ and output it.

$$rk_{ID\to PK} = \langle rk_1, rk_2\rangle = \left\langle PK_R^{\alpha}\left(g_1^{ID}h\right)^t, PK^t\right\rangle$$

$\mathbf{ReEnc_{PRO}}(\mathbf{ID}, \mathbf{rk_{ID\to PKE}}, \mathbf{params}, \mathbf{C_{IBE}})$. Given a delegator's identity ID, a re-encryption key $rk_{ID\to PK} = \langle rk_1, rk_2\rangle$, BB-IBE public parameter *params* and an IBE ciphertext C_{IBE} as input, the proxy re-encrypts and outputs a PKE ciphertext C_{PKE} or $\perp$.

1. Extract $C_{IBE} = \langle C_1, C_2, C_3\rangle$
2. Compute $v_1 = \hat{e}(C_1, g_1^{ID}h), v_2 = \hat{e}(C_2, g)$. If $v_1 \neq v_2$ then output $\perp$ and halt. Note that, correct input values can transform as follow:

$$\hat{e}(C_1, g_1^{ID}h) = \hat{e}(g^r, g_1^{ID}h) = \hat{e}(C_2, g)$$

3. Compute C_{PKE} and output it.

$$\bar{C}_{PKE} = \langle \bar{X}, \bar{Y}\rangle = \langle \hat{e}(rk_1, C_1), C_3 \cdot \hat{e}(rk_2, C_2)\rangle$$

The delegatee can decrypt this re-encrypt result $\bar{C}_{PKE}$ using his own secret key SK with same PKE decryption algorithm $\mathbf{Dec_{PKE}}(\mathbf{SK}, \mathbf{C_{PKE}}, \mathbf{params})$.

4.4 Security of IBE-PKE-PRE

Theorem 4.1. *Suppose that the (k, t, ε)-dBDH assumption holds in $(\mathbb{G}, \mathbb{G}_1)$. Then, the IBE-PKE-PRE is (k, t', q, ε)-IND-sPr-CPA secure against a $(TYPE = IBE)$ adversary for any (q, k, ε) and $t' < t - \Theta(\tau q)$, where τ denotes a maximum time for exponentiation in $\mathbb{G}, \mathbb{G}_1$.*

Proof. Let $\mathcal{A}_{IBE}$ be a t-time $(TYPE = IBE)$ adversary against the IBE-PKE-PRE. We construct an adversary $\mathcal{B}_{IBE}$ which can solve the dBDH problem in $\mathbb{G}$ by using $\mathcal{A}_{IBE}$. The $\mathcal{B}_{IBE}$ is given an input $\langle g, \Gamma_1, \Gamma_2, \Gamma_3, T\rangle = \left\langle g, g^a, g^b, g^c, T\right\rangle$, and distinguishes T is $\hat{e}(g, g)^{abc}$ or $T \in_R \mathbb{G}_1$. $\mathcal{B}_{IBE}$ works as follows:

Initialisation. $\mathcal{A}_{IBE}$ outputs an identity ID^* and notifies $\mathcal{B}_{IBE}$. $\mathcal{B}_{IBE}$ generates four blank lists to write down a query and answer pairs for every queries.

ISKL (IBE Secret Key List): Record the tuple $\langle ID_i, sk_{ID_i}\rangle$, where ID_i is an identity and an IBE secret key sk_{ID_i} corresponding to ID_i.

PPKL (PKE Public Key List): Record the tuple $\left\langle PKE_j, PK_j, PK_{R_j},\right\rangle$, where PKE_j is a PKE user identity and PK_j and PK_{R_j} are a public key and public key for re-encryption corresponding to PKE user identity PKE_j.

PSKL (PKE Secret Key List): Record the tuple $\left\langle PKE_j, SK_j, mark\right\rangle$, where PKE_j is a PKE user identity, SK_j are PKE secret key corresponding to PKE user identity PKE_j and mark keeps a flag that PKE user PKE_j is a *honest* party or *corrupted* party by $\mathcal{A}_{IBE}$.

REKL (Re-Encryption Key List): Record the tuple $\left\langle ID_i, PKE_j, rk_{ID_i\to PKE_j}, t_{i,j}\right\rangle$, where ID_i is an identity, PKE_j is a PKE user identity, $rk_{ID_i\to PKE_j}$ is a re-encryption key converts IBE ciphertext to PKE ciphertext and $t_{i,j}$ is a random number used for generating a re-encryption key.

Setup. The $\mathcal{B}_{IBE}$ generates a random number $z \in_R \mathbb{Z}_p^*$ and sets $g_1 = \Gamma_1$, $g_2 = \Gamma_2$, $h = g_1^{-ID^*}g^z$. $\mathcal{B}_{IBE}$ provides public parameters $params = \langle g, g_1, g_2, h\rangle$ to $\mathcal{A}_{IBE}$. Under these conditions, the master key value is g^{ab} which $\mathcal{B}_{IBE}$ cannot compute.

$\mathcal{B}_{IBE}$ generates random numbers $x_j \in_R \mathbb{Z}_p^*$ $(0 \leq j \leq l)$ where l denotes the number of PKE users, and sets the PKE public key and secret key as follows:

- If the PKE user PKE_j is a *corrupted* party by $\mathcal{A}_{IBE}$, sets the PKE public key as $PK_j = g^{x_j}$, the PKE public key for re-encryption as $PK_{R_j} = \Gamma_2^{1/x_j}$ and the secret key as $SK_j = x_j$. $\mathcal{B}_{IBE}$ stores $\left\langle PKE_j, PK_j, PK_{R_j}, SK_j\right\rangle$ to *PPKL* and *PSKL*, and sets the mark as *corrupted*.
- If the PKE user PKE_j is a *honest* party, sets the PKE public key as $PK_j = \Gamma_2^{x_j}$, the PKE public key for re-encryption as $PK_{R_j} = g^{1/x_j}$.
  Under this condition, PKE secret key value is $SK_j = bx_j$ where $\mathcal{B}_{IBE}$ cannot compute, however $\mathcal{B}_{IBE}$ can reject the query of this value. $\mathcal{B}_{IBE}$ stores the secret key as $SK_j = x_j$ as a substitute for computing re-encryption key values.

$\mathcal{B}_{IBE}$ stores $\langle PKE_j, PK_j, PK_{R_j}, SK_j \rangle$ to $PPKL$ and $PSKL$ and sets the mark as *honest*.

$\mathcal{B}_{IBE}$ gives $PPKL$ to $\mathcal{A}_{IBE}$.

Phase 1. $\mathcal{A}_{IBE}$ adaptively queries $\mathcal{B}_{IBE}$, and $\mathcal{B}_{IBE}$ responds as follows:

Extract($\mathbf{ID_i}$). $\mathcal{A}_{IBE}$ queries the IBE user's secret key sk_{ID_i} with an identity ID_i, then $\mathcal{B}_{IBE}$ generates a random number $u_i \in_R \mathbb{Z}_p^*$ and computes sk_{ID_i}.
If $ID_i = ID^*$, $\mathcal{B}_{IBE}$ rejects the query. Otherwise, $\mathcal{B}_{IBE}$ computes $sk_{ID_i} = \langle d_1, d_2 \rangle$ as follows:

$$\begin{aligned} d_1 &= g_2^{\frac{-z}{(ID_i - ID^*)}} \left(g_1^{(ID_i - ID^*)} g^z \right)^{u_i}, \\ d_2 &= g_2^{\frac{-1}{(ID_i - ID^*)}} g^{u_i}. \end{aligned}$$

$\mathcal{B}_{IBE}$ writes a request and a response to $ISKL$ and answers sk_{ID_i} to $\mathcal{A}_{IBE}$.

Extract$_{\mathbf{PKE}}$($\mathbf{PKE_j}$). $\mathcal{A}_{IBE}$ queries the PKE user's secret key SK_j with a PKE user's identity PKE_j, then $\mathcal{B}_{IBE}$ searches the $PSKL$ to retrieve PKE user's secret key SK_j
If PKE_j marked as *honest*, then $\mathcal{B}_{IBE}$. rejects, otherwise (PKE_j marked as *corrupted*) $\mathcal{B}_{IBE}$ answers SK_j retrieved from $PSKL$.

Extract$_{\mathbf{IBE \to PKE}}$($\mathbf{ID_i}, \mathbf{PKE_j}$). $\mathcal{A}_{IBE}$ queries the re-encryption key $rk_{ID_i \to PKE_j}$ which can converts ciphertexts from an identity ID_i to PKE_j, then $\mathcal{B}_{IBE}$ searches $PSKL$ to retrieve PKE_j record.

1. If $ID_i = ID^*$ and PKE_j marked as *corrupted*, then $\mathcal{B}_{IBE}$ rejects.
2. If $ID_i = ID^*$ and PKE_j is a *honest* party, then $\mathcal{B}_{IBE}$ generates random number $t_{*,j} \in_R \mathbb{Z}_p^*$ and computes $rk^{honest}_{ID^* \to PKE_j} = \langle rk_1^*, rk_2^* \rangle$ as follows:
$$\begin{aligned} rk_1^* &= g_1^{1/SK_j} (g^z)^{t_{i,j}}, \\ rk_2^* &= g_2^{t_{i,j} SK_j}. \end{aligned}$$
3. If $ID_i \neq ID^*$ and PKE_j marked as *corrupted*, $\mathcal{B}_{IBE}$ generates random number $t_{i,j} \in_R \mathbb{Z}_p^*$ and computes $rk^{corrupted}_{ID_i \to PKE_j} = \langle rk_1^c, rk_2^c \rangle$ as follows:
$$\begin{aligned} rk_1^c &= g_2^{\frac{-z}{SK_j (ID_i - ID*)}} \left(g_1^{(ID_i - ID*)} g^z \right)^{t_{i,j}}, \\ rk_2^c &= g_2^{\frac{-1}{ID_i - ID*}} g^{t_{i,j} SK_j}. \end{aligned}$$
4. If $ID_i \neq ID^*$ and PKE_j marked as *honest*, then $\mathcal{B}_{IBE}$ generates random number $t_{i,j} \in_R \mathbb{Z}_p^*$ and computes $rk^{honest}_{ID_i \to PKE_j} = \langle rk_1^h, rk_2^h \rangle$ as follows:
$$\begin{aligned} rk_1^h &= g^{\frac{-z}{SK_j (ID_i - ID*)}} \left(g_1^{ID_i - ID*} g^z \right)^{t_{i,j}}, \\ rk_2^h &= g_2^{\frac{-1}{ID_i - ID*}} g_2^{t_{i,j} SK_j}. \end{aligned}$$

$\mathcal{B}_{IBE}$ writes a request and a response pair to $REKL$, and answers $rk_{ID_i \to PKE_j}$ to $\mathcal{A}_{IBE}$.

Challenge. $\mathcal{A}_{IBE}$ outputs two equal length plaintexts M_0, M_1 and sends them to $\mathcal{B}_{IBE}$. $\mathcal{B}_{IBE}$ selects $d (\in_R \{0,1\})$ and encrypts M_d. $\mathcal{B}_{IBE}$ computes an IBE ciphertext C^*_{IBE} as follows:

$$C^*_{IBE} = \langle C_1^*, C_2^*, C_3^* \rangle = \langle \Gamma_3, (\Gamma_3)^z, M_d \cdot T \rangle$$

$\mathcal{B}_{IBE}$ sends C^*_{IBE} to $\mathcal{A}_{IBE}$. Note that, if $T = \hat{e}(g,g)^{abc}$, C^*_{IBE} is a correct IBE ciphertext of M_d under an identity ID^*.

Phase 2. $\mathcal{B}_{IBE}$ answers $\mathcal{A}_{IBE}$'s queries in same manner of **Phase 1**.

Solve. Finally, $\mathcal{A}_{IBE}$ outputs a guess result $d' \in \{0,1\}$. If $d' = d$, then $\mathcal{B}_{IBE}$ judges $T = \hat{e}(g,g)^{abc}$ and outputs 1; otherwise, $\mathcal{B}_{IBE}$ judges $T \in_R \mathbb{G}_1$ and outputs 0.

We claim that in the above simulation answers of $\mathcal{B}_{IBE}$ are correctly distributed, and $\mathcal{A}_{IBE}$ cannot distinguish our simulation from the real-world interaction.

Furthermore, $Adv^{dBDH}_{\mathcal{A}} = Adv^{S}_{\mathcal{A}_{IBE}}$, because $\mathcal{B}_{IBE}$ does not abort during the above simulation.

In the above simulation, maximum computation cost of the queries is at most polynomial time exponentiation, hence $t' < t - \Theta(\tau q)$. Therefor, the IBE-PKE-PRE is (k, t', q, ε)-IND-sPr-CPA secure against against an $(TYPE = IBE)$ adversary. □

Theorem 4.2. *Suppose that the (k,t,ε)-dBDH assumption holds in $(\mathbb{G}, \mathbb{G}_1)$. Then, the IBE-PKE-PRE is (k, t', q, ε)-IND-sPr-CPA secure against a $(TYPE = PKE)$ adversary for any (q, k, ε) and $t' < t - \Theta(\tau q)$ where τ denotes a maximum time for exponentiation in $\mathbb{G}, \mathbb{G}_1$.*

Proof. Let $\mathcal{A}_{PKE}$ be a t-time $(TYPE = PKE)$ adversary against the IBE-PKE-PRE. We construct an adversary $\mathcal{B}_{PKE}$ which can solve dBDH problem in $\mathbb{G}$, by using $\mathcal{A}_{PKE}$. The $\mathcal{B}_{PKE}$ is given an input $\langle g, \Gamma_1, \Gamma_2, \Gamma_3, T \rangle = \langle g, g^a, g^b, g^c, T \rangle$, and distinguishes T is $\hat{e}(g,g)^{abc}$ or $T \in_R \mathbb{G}_1$. $\mathcal{B}_{PKE}$ works as follows:

Initialisation. $\mathcal{B}_{PKE}$ generates four blank lists to write down a query and answer pairs for every queries, same as $(TYPE = IBE)$ proof.

Setup. The $\mathcal{B}_{PKE}$ generates a random number $w \in_R \mathbb{Z}_p^*$ and sets $g_1 = g^w$, $g_2 = \Gamma_2$, pick a random element h in $\mathbb{G}$. $\mathcal{B}_{PKE}$ provides public parameters

$params = \langle g, g_1, g_2, h \rangle$ to $\mathcal{A}_{PKE}$. Under these conditions, the master key value is g_2^w which $\mathcal{B}_{PKE}$ can compute.

$\mathcal{B}_{PKE}$ generates PKE user's key pairs and stores $PPKL$ and $PSKL$ same as $(TYPE = IBE)$ proof. $\mathcal{B}_{PKE}$ gives $PPKL$ to $\mathcal{A}_{PKE}$.

Phase 1. $\mathcal{A}_{PKE}$ adaptively queries $\mathcal{B}_{PKE}$, and $\mathcal{B}_{PKE}$ responds as follows:

$\mathbf{Extract_{IBE}(ID_i)}$. $\mathcal{A}_{PKE}$ queries the IBE user's secret key sk_{ID_i} with an identity ID_i, then $\mathcal{B}_{PKE}$ generates a random number $u_i \in_R \mathbb{Z}_p^*$ and computes $sk_{ID_i} = \langle d_1, d_2 \rangle$.

$$\begin{aligned} d_1 &= g_2^w \left(g_1{}^{ID_i} h\right)^{u_i}, \\ d_2 &= g^{u_i}. \end{aligned}$$

$\mathcal{B}_{PKE}$ writes a request and a response to $ISKL$ and answers sk_{ID_i} to $\mathcal{A}_{PKE}$.

$\mathbf{Extract_{PKE}(PKE_j)}$. $\mathcal{A}_{PKE}$ queries the PKE user's secret key SK_j with a PKE user's identity PKE_j, then $\mathcal{B}_{PKE}$ searches the $PSKL$ to retrieve PKE user's secret key SK_j

If PKE_j marked as *honest*, then $\mathcal{B}_{PKE}$ rejects, otherwise (PKE_j marked as *corrupted*) $\mathcal{B}_{PKE}$ answers SK_j retrieved from $PSKL$.

$\mathbf{Extract_{IBE \to PKE}(ID_i, PKE_j)}$. $\mathcal{A}_{PKE}$ queries the re-encryption key $rk_{ID_i \to PKE_j}$, which can convert ciphertexts from an identity ID_i to PKE_j, then $\mathcal{B}_{PKE}$ searches $PSKL$ to retrieve PKE_j record. $\mathcal{B}_{PKE}$ generates random number $t_{i,j} \in_R \mathbb{Z}_p^*$ and computes $rk_{ID_i \to PKE_j}$.

1. If PKE_j marked as *honest*, $\mathcal{B}_{PKE}$ computes $rk^{honest}_{ID_i \to PKE_j} = \langle rk_1^h, rk_2^h \rangle$ as follows:

$$\begin{aligned} rk_1^h &= g^{w/SK_j} \left(g_1^{ID_i} h\right)^{t_{i,j}}. \\ rk_2^h &= g_2{}^{t_{i,j} SK_j}, \end{aligned}$$

2. If PKE_j marked as *corrupted*, $\mathcal{B}_{PKE}$ computes $rk^{corrupted}_{ID_i \to PKE_j} = \langle rk_1^c, rk_2^c \rangle$ as follows:

$$\begin{aligned} rk_1^c &= g_2{}^{w/SK_j} \left(g_1^{ID_i} h\right)^{t_{i,j}}, \\ rk_2^c &= g^{t_{i,j} SK_j}. \end{aligned}$$

$\mathcal{B}_{PKE}$ writes a request and a response to $REKL$, and answers $rk_{ID_i \to PKE_j}$ to $\mathcal{A}_{PKE}$.

Challenge. $\mathcal{A}_{PKE}$ outputs two equal length plaintexts M_0, M_1 and selects target PKE user identity PKE^* in *honest* party and sends them to $\mathcal{B}_{PKE}$. $\mathcal{B}_{PKE}$ selects $d (\in_R \{0,1\})$ and encrypts M_d.

$\mathcal{B}_{PKE}$ retrieve selected PKE user's secret key $SK^* = x^*$ from $PSKL$ and computes a PKE ciphertext C^*_{PKE} as follows:

$$C^*_{PKE} = \langle X^*, Y^* \rangle = \left\langle \hat{e}(\Gamma_1, \Gamma_3)^{1/x^*}, M_d \cdot T \right\rangle$$

$\mathcal{B}_{PKE}$ sends C^*_{PKE} to $\mathcal{A}_{PKE}$. Note that, if $T = \hat{e}(g,g)^{abc}$, C^*_{PKE} is a correct PKE ciphertext of M_d under a PKE user identity PKE^*.

Phase 2. $\mathcal{B}_{PKE}$ answers $\mathcal{A}_{PKE}$'s queries in same manner of **Phase1**.

Solve. Finally, $\mathcal{A}_{PKE}$ outputs a guess result $d' \in \{0,1\}$. If $d' = d$, then $\mathcal{B}_{PKE}$ judges $T = \hat{e}(g,g)^{abc}$ and output 1; otherwise, $\mathcal{B}_{PKE}$ judges $T \in_R \mathbb{G}_1$ and outputs 0.

We claim that in the above simulation answers of $\mathcal{B}_{PKE}$ are correctly distributed, and $\mathcal{A}_{PKE}$ cannot distinguish our simulation from the real-world interaction.

Furthermore, $Adv^{dBDH}_{\mathcal{A}} = Adv^{S}_{\mathcal{A}_{PKE}}$, because $\mathcal{B}_{PKE}$ does not abort during the above simulation.

In the above simulation, maximum computation cost of the queries is at most polynomial time exponentiation, hence $t' < t - \Theta(\tau q)$. Therefor, the IBE-PKE-PRE is (k, t', q, ε)-IND-sPr-CPA secure against against an $(TYPE = PKE)$ adversary.

Remark 4.1. *We can simulate the game of Theorem 4.2 without simulating IBE secret key queries* $\mathbf{Extract_{IBE}(ID_i)}$, *re-encryption key queries* $\mathbf{Extract_{IBE \to PKE}(ID_i, PKE_j)}$, *and public keys for re-encryption* PK_{R_j}. *This implies that we can proof PKE scheme Chosen Plaintext secure under the dBDH assumption.*

□

5 CONCLUSIONS

In this paper, we propose a efficient [IBE-PKE]-type proxy re-encryption scheme which the size of the re-encrypted ciphertext is optimal and delegatee does not aware of existence of the proxy. We define the security notation and prove selective-ID secure based on dBDH assumption in the standard model against chosen plaintext attack. Furthermore our scheme might be possible to extend full-ID secure using IBE proposed in (B.Waters, 2005).

Green and Ateniese (M.Green and G.Ateniese, 2007) proposed the semantically secure Identity-Based proxy re-encryption scheme and constructed CCA-secure scheme applying CHK conversion technique (R.Canetti et al., 2004) to their CPA-secure scheme. It might be able to construct the CCA-secure [IBE-PKE]-type proxy re-encryption scheme by using same technique to our CPA-secure scheme. It will be appeared in the full version.

REFERENCES

B.Waters (2005). Efficient identity-based encryption without random oracles. In *In Proceedings of Eurocrypt '05, volume 3494 of LNCS*, pages 114–127. Springer-Verlag.

D.Boneh and X.Boyen (2004). Efficient selectiveid secure identity based encryption without random oracle. In *In Advances in Cryptology - EUROCRYPT'04, volume 3027 of LNCS*, pages 223–238. Springer-Verlag.

G.Ateniese, K.Fu, M.Green, and S.Hohenberger (2005). Improved proxy re-encryption schemes with applications to secure distributed storage. In *In Proceedings of the 12th Annual Network and Distributed System Security Symposium - NDSS'05*, pages 83–107.

L.Zbou, M.A.Marsh, F.B.Schneider, and A.Redz (2004). Distributed blinding for elgamal reencryption. In *Technical Report 2004-1924*. Cornell Computer Science Department.

M.Blaze, G.Bleumer, and M.Strauss (1998). Divertible protocols and atomic proxy cryptography. In *In Advances in Cryptology - EUROCRYPT'98, volume 1403 of LNCS*, pages 127–144. Springer-Verlag.

M.Green and G.Ateniese (2007). Indentity-based proxy re-encryption. In *ACNS 2007, volume 4521 of LNCS*, pages 288–306. Springer-Verlag.

M.Jakobsson (1999). On quorum controlled asymmetric proxy re-encryption. In *In Proceedings of Public Key Cryptography - PKC'99, volume 1560 of LNCS*, pages 112–121. Springer-Verlag.

M.Mambo and E.Okamoto (1997). Proxy cryptosystems: Delegation of the power to decrypt ciphertexts. In *IEICE Trans. Fund. Electronics Communications and Computer Science E80-A/1*, pages 54–63. IEICE.

R.Canetti, S.Halevi, and J.Katz (2004). Chosen-ciphertext security from identity based encryption. In *In Proceedings of Eurocrypt '04, volume 3027 of LNCS*, pages 207–222. Springer-Verlag.

R.Canetti and S.Hohenberger (2007). Chosen-ciphertext secure proxy re-encryption. In *CCS '07: Proceedings of the 14th ACM conference on Computer and communications security*, pages 185–194. ACM.

T.Matsuo (2007). Proxy re-encryption systems for identity-based encryption. In *In Proceedings of Pairing-Based Cryptography - Pairing'07, volume 4575 of LNCS*, pages 247–267. Springer-Verlag.

Y.Dodis and A.Ivan (2003). Proxy cryptography revisited. In *In Proceedings of the 10th Annual Network and Distributed System Security Symposium- NDSS'03*.